

REMARKS/ARGUMENTS

1. Claims 1-10, 13, 15, 17-26, 29, 31, 33-42, 45, and 47 are Patentable Over the Cited Art

Claims 1-10, 13, 15, 17-26, 29, 31, 33-42, 45, and 47 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Putzolu (U.S. Pat. No. 6,681,243) in view of Flores (U.S. No. 5,630,069). This rejection is respectfully traversed.

Independent claims 1, 17, and 33 concern enabling access to a plurality of service engines, wherein each service engine enables access to service resources, and require: providing a plurality of service class implementations for service engines from different vendors, wherein each service class implementation provides an implementation of methods and objects from a same abstract service class; instantiating a service object for one service engine in response to at least one called method from one of the service class implementations, wherein the service object includes information on the service engine; receiving a method call from one service class implementation requesting information on service engine resources for one named service; and using the service object to access the requested information to return to the method call.

The Examiner cited col. 14, lines 42-67 and col. 15, lines 1-67 of Flores as teaching the claim requirement of providing a plurality of service class implementations for service engines from different vendors, wherein each service class implementation provides an implementation of methods and objects from a same abstract service class. (Third Office Action, pg. 3)
Applicants traverse.

The cited col. 14 discusses a Model-View-Controller (MCV) framework that provides a split of the different functions in a GUI application. Isolating the core application logic in the model makes the application more portable and the implementation extendible. The logical separation of the event handling in the Controller from the user interface in the View enables the application to be more easily ported to another GUI environment. The Model classes describe the business process and components in terms of a hierarchy of classes, the View classes draw the workflow map of the business process on different displays.

Although the cited col. 14 discusses logical separation of a GUI application, nowhere does the cited col. 14 teach or suggest providing service class implementations of a same abstract service class from different vendors as claimed. There is no mention or suggestion in the cited col. 14 of service class implementations of a service engine, enabling access to service resources,

from different vendors, where each implementation is of a same abstract service class. Instead, the cited col. 14 mentions a logical separation of the functions of a GUI application, not service class implementations of a same abstract service class from different vendors as claimed.

The cited col. 15 discusses different classes used to implement a user interface. For instance, an ActWfView class receives the menu and toolbar commands from the views and passes them to a window. A MapView class has a painter and controller components, where the controller component contains the menu, toolbar, mouse and keyboard interpreters. This class has methods to receive input from the users. The cited col. 15 further discusses various methods for a user interface, such as mouse methods, shapes, painting components to display images and areas. Further, the view classes get information from the model class, where the main interface for the model classes is a series of functions to manage attributes of the class.

Nowhere does the cited col. 15 anywhere teach, suggest or mention providing service class implementations of a same abstract service class from different vendors as claimed. The cited col. 15 discusses classes and methods for a user interface. There is no teaching or mention in the cited col. 15 of service class implementations from a same abstract service class provided by different vendors. Instead, the cited col. 15 discusses classes and methods to implement a user interface.

The Examiner further cited col. 1, lines 7-10 when finding that Flores teaches providing service class implementation for service engines from different vendors. (Third Office Action, pgs. 3-4) Applicants traverse.

The cited col. 1 mentions that the method provides consultants, business process analysts, and developers with a unified tool with which to conduct business process analysis and design. Nowhere does this cited col. 1 or other cited sections of Flores anywhere teach or suggest providing service class implementation for service engines from different vendors. Instead, the cited col. 1 discusses a tool to allow people to analyze business processes.

The Examiner cited the Abstract and col. 7, line 41 to col. 8, line 10 of Putzolu as teaching the claim requirement that each service class implementation provides an implementation of methods and objects from a same abstract service class. (Third Office Action, pg. 2) Applicants traverse on the grounds that the cited art does not teach the specific claimed combination of service class implementations of a same abstract service class from different vendors as claimed.

The cited cols. 7-8 mention providing an interface to agents including services. Services are Java classes instantiated as objects that have methods that accept inputs from agents and allow agents to access resources. Each resource on a network is accessible by agents via a service, such as the hard disk drive on each device is accessible by a disk service type.

Although the cited cols. 7-8 mention service classes that allow agents to access resources, nowhere do the cited cols. 7-8 teach or suggest providing service class implementations of a same abstract service class from different vendors as claimed. In fact, cols. 7-8 teach away from this requirement because the cited cols. 7-8 mention that “[d]ifferent types of services provide access to underlying resources in a different manner and interface with agents in a different manner” and that “the actual service class used by an agent to access a service type on each device may be different”. Thus, cols. 7-8 do not teach service class implementations of methods and objects from a same abstract service class or from different vendors. Instead, the cited cols. 7-8 mention how services for different resources use different types of services and service classes, not implementations of the same abstract service class as claimed.

The cited Abstract discusses allowing agents to function on devices having resources and that agents may move from an environment on one device to an environment on another device. Nowhere does the cited Abstract teach or suggest providing service class implementations of a same abstract service class as well as implementations from different vendors as claimed.

Accordingly, Applicants submit that claims 1, 17, and 33 are patentable over the cited art because the cited combination does not teach or suggest all the claim requirements.

Claims 2-10, 13, 15, 18-26, 31, 34-42, and 47 are patentable over the cited art because they depend from one of claims 1, 17, and 33, which are patentable over the cited art for the reasons discussed above. Moreover, the below discussed dependent claims provide additional grounds of patentability over the cited art for the reasons discussed.

Claims 3, 19, and 35 depend from claims 1, 17, and 33 and further require that the service engines include workflow products from different vendors, wherein the workflow products comprise computer programs enabling implementation of a computer implemented workflow defining a series of processes to be performed by users at computers with respect to a computer implemented work item.

The Examiner cited all cols. 16, 17, and 18 of Flores as teaching the additional requirements of these claims. (Third Office Action, p. 4) Applicants traverse.

The cited col. 16 discusses a model class that implements workflow application logic, that describes the business process and its components in terms of classes. The cited col. 16 further discusses different classes and their attributes. The WfComponent class is an abstract class providing the base for all classes which represent components of a process.

The cited col. 17 discusses various workflow classes. A WfAnchor class is an abstract class which provides the meaning for origin/target objects as Workflows and conditional links. The WfWorkflow is a class that models the logical concept of the workflow, including customer performer, etc. Col. 18 discusses additional workflow related classes.

Although the cited cols. 16, 17, and 28 discuss classes that describe the business process, its component, and workflow concepts, there is no teaching or suggestion in the cited cols. 16, 17, and 18 of service engines including workflow products from different vendors.

Accordingly, the additional requirements of claims 3, 19, and 35 provide additional grounds of patentability over the cited art.

Claims 4, 20, and 36 depend from claims 3, 19, and 35 and further require that the workflow service class implementations from different vendors each includes methods and objects from a same abstract workflow service class specifying methods and objects to include in all workflow service class implementations.

The Examiner cited col. 19, lines 51-67 and col. 20, lines 1-67 of Flores as teaching the additional requirements of these claims. (Third Office Action, pg. 4) Applicants traverse.

The cited col. 19 mentions that the Model utilizes workflow rules determined by the workflow structure, such as the number of phases, roles, workflow acts, cycle times, positions in the phase line where a link can be drawn, etc. Nowhere does this cited col. 19 anywhere teach or suggest the claim requirement of workflow service class implementations from different vendors including methods and objects from a same abstract workflow service class specifying methods and objects to include in all workflow service class implementations.

The cited col. 20 mentions that an I/O module stores the model classes in map files. The view classes implement the user interface components to draw the model classes on the display. The cited col. 20 discusses view classes to display a map and draw the workflow symbols on the map, and classes to print the workflow symbols, and a shape class to derive the shapes to be drawn on the map.

Although the cited cols. 19 and 20 discuss classes for a workflow and classes to view and draw symbols for a workflow, there is still no teaching, suggestion or mention of the claim requirement of workflow service class implementations from different vendors including methods and objects from a same abstract workflow service class.

Accordingly, the additional requirements of claims 4, 20, and 36 provide additional grounds of patentability over the cited art.

Applicants further submit that the additional requirements of other of the dependent claims in combination with the base claims provide further grounds of distinction over the cited art.

2. Claims 11, 12, 14, 16, 27, 28, 30, 32, 43, 44, 46, and 48 are Patentable Over the Cited Art

The Examiner rejected claims 11, 12, 14, 16, 27, 28, 30, 32, 43, 44, 46, and 48 as obvious (35 U.S.C. §103(a)) over Putzolu in view of Flores and further in view of Wollrath (U.S. Patent No. 6,487,607). Applicants traverse.

Applicants submit that the cited claims 11, 12, 14, 16, 27, 28, 30, 32, 43, 44, 46, and 48 are patentable over the cited art because they depend from one of claims 1, 17, and 33, which are patentable over the cited art for the reasons discussed above. Moreover, the additional requirements of these claims in combination with the base claims provide further grounds of distinction over the cited art.

Conclusion

For all the above reasons, Applicants submit that the pending claims 1-48 are patentable over the art of record. Applicants have not added any claims. Nonetheless, should any additional fees be required, please charge Deposit Account No. 09-0460.

\\
\\
\\
\\
\\
\\

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: April 5, 2006

By: /David Victor/

David W. Victor
Registration No. 39,867

Please direct all correspondences to:

David Victor
Konrad Raynes & Victor, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984